IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:		Atty. Docket No.:	006020.00027
Philip Victor HARMAN et al.			
Serial No.:	TBA	Group Art Unit:	Unknown
Filed:	November 25, 2003	Examiner:	Unknown
For:	OPEN GL	Confirmation No.:	Unknown

CLAIM FOR PRIORITY UNDER 35 U.S.C. §119

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The benefit of the filing date of the following prior foreign application is hereby requested for the above-identified application and the priority provided under 35 U.S.C. §119 is hereby claimed: (a certified copy of the foreign application is enclosed herewith)

Country	Application Number	Date of Filing (day, month, year)
Australia	2002952872	25 November 2002

It is requested that the file of this application be marked to indicate that the requirements of 35 U.S.C. §119 have been fulfilled and that the Patent and Trademark Office kindly acknowledge receipt of these documents.

Respectfully submitted,
BANNER & WITCOFF, LTD.

Dated: November 25, 2003

By: Gary D. Fedorochko
Gary D. Fedorochko
Registration No. 35,509

Eleventh Floor
1001 G Street, N.W.
Washington, D.C. 20001-4597
(202) 824-3000

GDF:lab

AUSTRALIA

Patent Office
Canberra

I, JULIE BILLINGSLEY, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. 2002952872 for a patent by DYNAMIC DIGITAL DEPTH RESEARCH PTY LTD as filed on 25 November 2002.

WITNESS my hand this
Twenty-third day of May 2003

J. Billingsley

JULIE BILLINGSLEY
TEAM LEADER EXAMINATION
SUPPORT AND SALES

P/00/009 28/5/91
Regulation 3.2

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

Invention Title: **IMAGE GENERATION**

The invention is described in the following statement:

IMAGE GENERATION

FIELD OF THE INVENTION

The present invention is directed towards an improved technique for the generation of images for use with autostereoscopic displays. In particular the present invention relates to a method of automatically producing images derived from computer generated source images.

BACKGROUND OF THE INVENTION

A number of autostereoscopic displays are starting appear on the market.

Whilst some of these displays require a conventional stereoscopic pair of images others require multiple images in order to provide an autostereoscopic image. Such multiple image displays enable the viewer to retain an stereoscopic effect despite movement of their head.

Such displays generally require images comprising a number of views created from a number of laterally displaced cameras. Such views can be originated from real cameras or generated using computer graphic image (CGI) techniques.

In US patents 6,118,584 and 6,064,424, included here in full by reference, van Berkel describes an autostereoscopic display that requires seven views. In German patent PCT WO 01/56302 A1, included here in full by reference, Grasnik describes an autostereoscopic display that requires eight views.

Such displays are known to require multi-views or integer multi-views in order to display autostereoscopic images.

Given the commercial availability of such multi-view displays there is a corresponding requirement for suitable images or content.

Whilst such content can be obtained from a number of laterally displaced cameras this invention relates to the generation of images using CGI techniques.

CGI 3D models can be created and 2D views generated. A 2D view can be characterised by the location of a virtual camera. The x, y and z location of the virtual camera defines the resulting 2D image. Those skilled in the art will be familiar with CGI software packages that provide such a capability and 3D Studio Max by Discrete Inc. is an example of such a package.

The prior art also teaches that stereoscopic images, i.e. ind pendent left and right eye images may be rendered from a CGI 3D model. This is achieved by

rendering one image and then laterally displacing the virtual camera to produce a second image. Those skilled in the art will appreciate that the stereoscopic effect so produced is a function of the separation and toe-in of such virtual cameras.

In order to support multi-view screens it would therefore appear obvious to simply render multiple lateral views, one for each virtual camera location. This technique was proposed by van Berkel in a paper "Image Preparation for 3D-LCD" presented at the IS&T/SPIE Conference on Stereoscopic Displays and Applications X, San Jose, California, January 1999.

In this paper van Berkel proposes that image preparation is best undertaken at the graphics API (application programming interface) level and gives an example using the OpenGL API.

The current invention discloses a more efficient technique for developing images for multi-view autostereoscopic displays as well as a technique for improving the quality of images displayed.

OBJECT OF THE INVENTION

It is therefore an object of the present invention to provide an improved method for the real time generation of images, from CGI sources, suitable for use with multi-view autostereoscopic displays, and for improving the quality of such images.

SUMMARY OF THE INVENTION

With the above object in mind the present invention provides in one aspect a method of creating images suitable for use with a multi-view autostereoscopic display including the steps of:

1. A capturing means for intercepting 3D geometric primitives and associated characteristics passed between an application and an application programmers interface (API)

2. A view generation means for imaging said 3D geometric primitives and said associated characteristics from multiple distinct viewing positions

3. A mask calculation means for determining the relative contribution of each view based on the characteristics of an associated lenticular lens array

4. An accumulation means for combining said views with said masks to generate a composite 3D image

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates the principle of the invention.

Figure 2 illustrates how API calls are intercepted by a modified API

Figure 3 illustrates the principle of a slanted lenticular 3D-LCD following van Berkel.

DETAILED DESCRIPTION OF THE INVENTION

In order to be commercially successful multi-view autostereoscopic displays require to be supported with suitable content. It is desirable that such content be produced inexpensively and in large quantities.

Whilst it is possible to create original content for such displays using a number of laterally displaced video cameras it has been found in practice that such configurations are impractical. An alternative is to create images using CGI techniques and the prior art teaches how existing 3D graphics software, using multiple virtual cameras, can provide suitable images at low cost and complexity.

Many existing graphics applications use an application programming interface (API) in order to generalise the generation of computer images. Those skilled in the art will be familiar with OpenGL, DirectX, Direct3D and Starbase API's.

Such graphics applications include, although not limited to, games, simulations, architectural design, modelling and molecular design.

If access to the source code of such applications is available then it is possible to modify the code so as to produce an image suitable for directly displaying on a multi-view autostereoscopic display.

Such a technique is described in the document '4D-Vision Programming Manual' available from 4D-Vision, GmbH, Jena, Germany.

Such a process is non trivial and requires the close co-operation of the owners of the original source code. As such it is expected that in practice only a limited range of applications will be available for modification for use on a multi-view display.

It is known to those skilled in the art that personal computer graphics cards require software drivers in order to correctly interpolate an applications graphics commands and convert them into images. In general, each particular brand, make and model of graphics card requires a custom written software driver. It will be appreciated to those skilled in the art that it would be possible to modify the

source code of the driver so as to produce an image suitable for directly displaying on a multi-view autostereoscopic display.

Such a process has been demonstrated by 4D-Vision GmbH in the production of a DirectX driver, called mo3dst3d.exe, for their range of autostereoscopic screens.

It will be appreciated that in order to address the widest market possible then it is desirable that a large number of graphics cards are available to support the multi-view function. The modification of such drivers is non trivial and requires the close co-operation of the owners of the original source code. As such it is expected that in practice only a limited range of graphics cards will be available for modification for use with a multi-view display.

This invention discloses a technique where images suitable for display on a multi-view autostereoscopic screen can be generated without the need to modify either the original graphics application or the graphics card driver.

The basic principle of the invention is shown in Figure 1. Here a software application 1, communicates with an API 2 to a display driver 3. For purposes of explanation only the OpenGL API will be used, however, those skilled in the art will appreciate that the invention can be applied to other graphics API's.

OpenGL is a state machine which is modified by the execution of a series of commands by the client program. The client program sends a stream of commands to render the elements of the scene. This stream of commands is intercepted by the current invention in order to generate a 3D image. As described above, software applications use system libraries or APIs to access the functionality of the operating system. The software application searches for these libraries in a number of directories in the computer's file system. The current invention intercepts calls to system libraries by placing a modified library, with the same name and programming interface, in a directory that is searched before the directory containing the actual operating system's library. In an alternative embodiment, the operating system's library may be temporarily or permanently replaced by the modified library. In either case, the calls intended for the system library are received or intercepted by the modified library that may convert the original 2D signal into a 3D signal. This process is illustrated in figure 2 where calls from the software application 1 are intercepted by a modified library 2 for the

purposes of generating a 3D image. The modified signal is then passed on to the system API and subsequently to the display driver for rendering.

The requirement for generating multiple views is to render the scene multiple times using displaced camera positions. Without the benefit of modifying the source code the client or, in fact, knowing what the client program will do, technique is to identify replayable sequences of OpenGL commands using heuristics to identify valid sequences and to optimise the length of the replayable commands. The goal of length optimisation is to minimise the number of command flushing stages which incur a performance penalty.

The replayable command list is then used to render multiple views into OpenGL texture buffers, or, if that capability of the driver is not available, into portions of the final display buffer. Thus, over the command stream for the frame, the scene will be gradually drawn for each of the multiple views.

In a second embodiment this invention may be used to improve the image quality of multi-view autostereoscopic display systems using slanting optical systems. Such a display is described by van Berkel in US patents 6,118,584 and 6,064,424 and by Grasnik in German patent PCT WO 01/56302 A1.

Whilst van Berkel discloses the use of a slanting lenticular lens and Grasnik the use off a slanting wavelength selective filter it will be appreciated by those skilled in the art that the resulting optical effect, and need for multiple images, is the same.

Both van Berkel and Grasnik disclose the use of multiple views mapped in a specific manner to pixels on a flat panel display device, examples are given using LCD and Plasma displays.

In addition to the disclosure in van Berkel's patents, in a paper "Image Preparation for 3D-LCD" presented at the IS&T/SPIE Conference on Stereoscopic Displays and Applications X, San Jose, California, January 1999 he illustrates the use of a slating lenticular applied to an LCD display to form an autostereoscopic multi-view display.

Van Berkel's Figure 1 from this paper is reproduced as Figure 3 in this document. This figure shows the mapping of camera views, numbered 1 to 7, to pixels on the underlying LCD display.

For such a display system, either real or virtual cameras could be used to create the seven views and map them to the required pixels. It should be noted that both van Berkel and Grasnik disclose the use of integer views i.e. each pixel is directly mapped to one of N real or virtual cameras.

In van Berkel's SPIE paper he discloses an equation that can be used to calculate the view number *N* for each pixel *k,l* which can then be used to assign the appropriate image data to the pixel.

The equation is:

$$N = \frac{(k + k_{offset} - 3l\tan\alpha) \bmod X}{X} N_{tot}$$

where k is horizontal pixel index

k_{offset} is horizontal shift if lenticular lens array

$\alpha$is angle of lenticular lens array

X is views per lens

N_{tot} is total number of views

and N is the view number of each sub pixel k,l

Van Berkel goes on to teach that for arbitrary values of α and X and finite values of N_{tot}, N will not be an integer. In that case he teaches to take the nearest integer input image to provide the information for a given pixel.

However, it will be appreciated that using the nearest integer approach will in fact result in an inferior image. The effect of using the nearest integer rather than the exact view is illustrated in the following table:

Nearest Integer View

1	3	5	7	2	4	6	1
7	2	4	6	1	3	5	7
6	1	3	5	7	2	4	6
5	7	2	4	6	1	3	5

Intermediate View

0.572	2.572	4.572	6.572	1.572	3.572	5.572	0.572

6.568	1.568	3.568	5.568	0.568	2.568	4.568	6.568
5.563	0.563	2.563	4.563	6.563	1.563	3.563	5.563
4.559	6.559	1.559	3.559	5.559	0.559	2.559	4.559

With N_{tot}=7, X=3.5, α=9.5 and k_{offset}=10.8 (parameters selected to illustrate potential difference between integer and fractional views)

It is appreciated why van Berkel teaches the use of integer views since the use of intermediate views would, in the above example, require 28 intermediate views as opposed to seven integer views.

Such intermediate views could be created using real or virtual cameras. In the former this is considered impractical and in the latter the overhead of rending 28 views would place a significant load on the computer and potentially adversely effect the response time.

It will also be appreciated by those skilled in the art that the exact fractional views are dependant on the value of α and will therefore most likely vary between different screens. Hence, in practice, it is desirable to provide a solution that enables the exact intermediate views to be generated to match the optical characteristics of a particular display.

As explained above, for CGI applications, the prior art teaches to create intermediate views by positioning virtual cameras and rendering all the necessary views. Whilst this would produce the desired multi-view image it does place a significant additional load on the computer. In particular where the computer is being used to play a game such an additional processing load may slow the computer so as to adversely affect the game-play.

It is therefore desirable to provided a generic solution to the problem of providing the exact intermediate views with minimal computational overheads. Our invention teaches this by generating a number of equally spaced i.e. integer views, for example purposes only we will use seven views, and produce the exact intermediate views using synthesis techniques.

In a preferred embodiment generation of intermediate views includes two steps. The first is to calculate modulation mask textures that are used to represent the fractional proportion of each actual view for each pixel (red, green and blue) component. Thus there is one mask for each exact view.

An example of a proportion may be that the red component of a particular pixel is best approximated by 30% of view 3 and 70% of view 4. The masks are generated specifically to suit the particular optical characteristics of the display in use.

The mask values are calculated using the characteristics of the screen. Essentially it is necessary to calculate two variables, the number of views per colour component V_c and the number views per image row, V_r.

$$V_c = \frac{N_{tot}/3}{P_\mu}$$

$$V_r = \frac{N_{tot}\tan(\alpha)}{P_\mu}$$

Where P_μ is the horizontal component of the lenticular pitch and is derived from:

$$P_\mu = P\sqrt{1+\tan(\alpha)^2}$$

where P is the lenticular pitch and α is the angle of the lenticular lens and N_{tot} is the total number of distinct views.

To form a composite 3D image for a lenticular LCD the image is traversed in a raster scan. For the first position in the raster scan the view is initialised to an arbitrary view number. For each subsequent red, green or blue colour component in the same row of the raster scan the previous view is incremented by the value of V_c. As V_c may be a fractional value (for example, 2.63) the view calculated is fractional. Similarly, as the raster scan advances to a new row the view is incremented by Vr.

The determination of the modulation mask value from the fractional view is a simple weighted average of the closest 2 views. For any given fractional view V_f the mask values can be calculated using the following procedure:

1. Set all masks to zero.
2. *mask*(floor(V_f))=ceil(V_f)-V_f
3. *mask*(ceil(V_f))=V_f-floor(V_f)

For example, if we have a fractional view V_f=3.8 then for that specific pixel *mask*(3)=0.2 and *mask*(4)=0.8 and all other masks are equal to zero.

The sum of all proportions for all masks for an individual pixel component will be 100%.

The second step of intermediate view generation is to compose all rendered exact views to generate the final image for display. This is done by applying a simple summation function for each pixel component (red, green ,blue) as follows:

$$value = \sum_{i=1}^{N_{mu}} view(i) mask(i)$$

Practically, this is implemented using a texture cascade where the first texture stage is the exact view and the second texture stage the mask view. Then, a modulation operator is specified for the textures. A rectangle is drawn the size of the screen with textures indexed appropriately.

The frame buffer is set to accumulate (after being initially cleared) and all views iterated with their corresponding masks using the modulation procedure. Thus, this achieves the required summation function as described above.

Whilst the method and apparatus of the present invention has been summarised and explained by illustrative application it will be appreciated by those skilled in the art that many widely varying embodiments and applications are within the teaching and scope of the present invention, and that the examples presented herein are by way of illustration only and should not be construed as limiting the scope of this invention.

DATED this 25th day of November 2002

DYNAMIC DIGITAL DEPTH RESEARCH AUSTRALIA PTY LTD

WATERMARK PATENT & TRADE MARK ATTORNEYS
GPO BOX 2512
PERTH WA 6001
AUSTRALIA

FIGURES

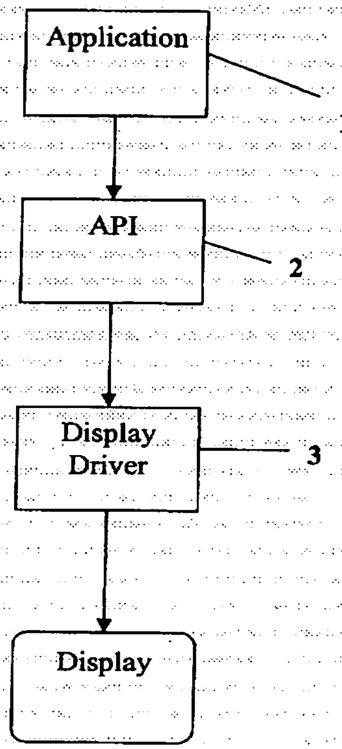

Figure 1

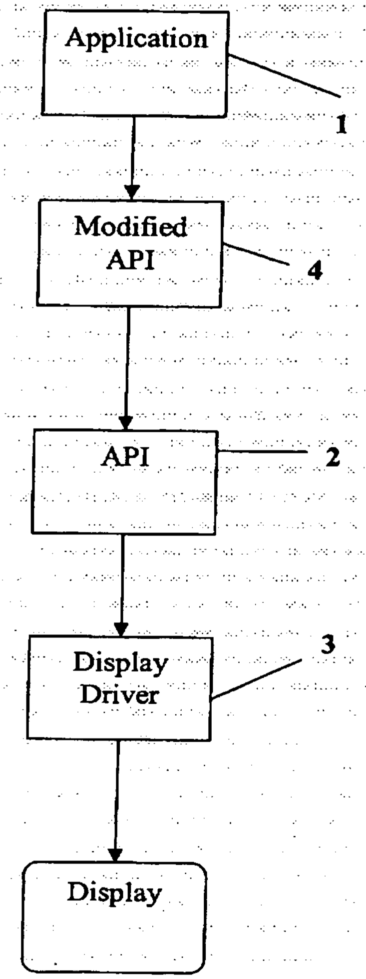

Figure 2

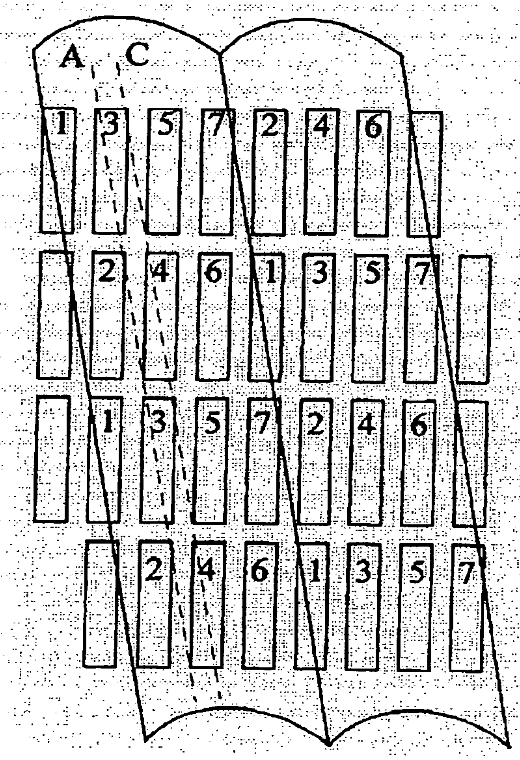

Figure 3: Slanted lenticular 3D-LCD (following van Berkel)